AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) An embedded disk controller having a servo controller, comprising:

a servo controller interface with a speed matching module and a pipeline control module such that at least two processors including first and second processors share memory mapped registers without conflicts.

2. (Currently Amended) The controller of Claim 1, where ~~one~~ the first processor operates at a first frequency and ~~a~~ the second processor operates at the second frequency.

3. (Original) The controller of Claim 1, where the servo-controller and the servo controller interface operate in same or different frequency domains.

4. (Original) The controller of Claim 1, the speed matching module ensures communication without inserting wait states in a servo controller interface clock domain for write access to the servo controller.

5. (Currently Amended) The controller of Claim 1, where there ~~is~~ are no read conflicts between the first and second processor.

6. (Original) The controller of Claim 1, provides a hardware mechanism for indivisible register access to the first or second processor.

7. (Currently Amended) The controller of Claim 6, where the hardware mechanism includes a ~~hard~~ semaphore.

8. (Cancelled)

9. (Currently Amended) The controller of Claim 1, where the pipeline control module resolves conflict between ~~the~~ first and second processor transactions of the first and second processors, respectively.

10. (Previously Presented) The controller of Claim 1, where the first and second processor communicate with the servo controller via two separate buses.

11. (Previously Presented) The controller of Claim 1, where if there is a write conflict between the first and second processor, pipeline control module holds write access to the second processor.

12. (Previously Presented) The controller of Claim 6, where the hardware mechanism is a semaphore register.

13. (Currently Amended) A system for reading and writing data to a storage medium, comprising:

an embedded disk controller having a servo controller interface module that includes a speed matching module and a pipeline control module such that at least two processors including first and second processors share memory mapped registers without conflicts.

14. (Currently Amended) The system of Claim 13, where ~~one~~ the first processor operates at a first frequency and ~~a~~ the second processor operates at the second frequency.

15. (Previously Presented) The system of Claim 13, where the servo-controller and the servo controller interface operate in same or different frequency domains.

16. (Previously Presented) The system of Claim 13, the speed matching module ensures communication without inserting wait states in a servo controller interface clock domain for write access to the servo controller.

17. (Previously Presented) The system of Claim 13, where there ~~is~~ are no read conflicts between the first and second processor.

18. (Previously Presented) The system of Claim 13, provides a hardware mechanism for indivisible register access to the first or second processor.

19. (Currently Amended) The system of Claim 18, where the hardware mechanism includes a ~~hard~~ semaphore.

20. (Cancelled)

21. (Currently Amended) The system of Claim 13, where the pipeline control module resolves conflict between ~~the~~ first and second processor transactions of the first and second processors.

22. (Previously Presented) The system of Claim 13, where the first and second processor communicate with the servo controller via two separate buses.

23. (Previously Presented) The controller system of Claim 13, where if there is a write conflict between the first and second processor, pipeline control module holds write access to the second processor.

24. (Previously Presented) The system of Claim 18, where the hardware mechanism is a semaphore register.

25. (Previously Presented) A servo controller interface for a disk controller, comprising:

a first interface for communicating with a first processor over a first bus; and

a second interface for communicating with a second processor over a second bus,

wherein the servo controller interface selectively grants one of the first and second processors access to a servo controller.

26. (Previously Presented) The servo controller interface of claim 25 wherein the first processor operates at a first frequency and the second processor operates at a second frequency.

27. (Previously Presented) The servo controller interface of claim 25 further comprising a speed matching module that resolves conflicts between at least first and second clock domains.

28. (Previously Presented) The servo controller interface of claim 27 wherein the speed matching module transitions servo controller accesses from one of the first and second clock domains to the other of the first and second clock domains.

29. (Previously Presented) The servo controller interface of claim 25 wherein the first and second processors share memory mapped registers within the servo controller.

30. (Previously Presented) The servo controller interface of claim 27 wherein the speed matching module does not insert wait states in a clock domain of the servo controller interface during write access to the servo controller.

31. (Previously Presented) The servo controller interface of claim 25 further comprising a pipeline control module that resolves transaction conflicts between the first processor and the second processor.

32. (Previously Presented) The servo controller interface of claim 25 wherein the servo controller interface delays a write access for one of the first and second processors during write conflicts between the first and second processors.

33. (Previously Presented) A servo controller interface for a disk controller, comprising:

first interface means for communicating with a first processor over a first bus; and

second interface means for communicating with a second processor over a second bus, wherein the servo controller interface selectively grants one of the first and second processors access to a servo controller.

34. (Previously Presented) The servo controller interface of claim 33 wherein the first processor operates at a first frequency and the second processor operates at a second frequency.

35. (Previously Presented) The servo controller interface of claim 33 further comprising speed matching means for resolving conflicts between at least first and second clock domains.

36. (Previously Presented) The servo controller interface of claim 35 wherein the speed matching means transitions servo controller accesses from one of the first and second clock domains to the other of the first and second clock domains.

37. (Previously Presented) The servo controller interface of claim 33 wherein the first and second processors share memory mapped registers within the servo controller.

38. (Previously Presented) The servo controller interface of claim 35 wherein the speed matching module does not insert wait states in a clock domain of the servo controller interface during write access to the servo controller.

39. (Previously Presented) The servo controller interface of claim 33 further comprising pipeline control means for resolving transaction conflicts between the first processor and the second processor.

40. (Previously Presented) The servo controller interface of claim 33 wherein the servo controller interface delays a write access for one of the first and second processors during write conflicts between the first and second processors.

41. (Previously Presented) A method for operating a servo controller interface having a first interface and a second interface, comprising:

communicating with a first processor over a first bus using the first interface;

communicating with a second processor over a second bus using the second interface; and

selectively granting one of the first and second processors access to a servo controller with the servo controller interface.

42. (Previously Presented) The servo method of claim 41 wherein the first processor operates at a first frequency and the second processor operates at a second frequency.

43. (Previously Presented) The method of claim 41 further comprising resolving conflicts between at least first and second clock domains at a speed matching module.

44. (Previously Presented) The method of claim 43 further comprising transitioning servo controller accesses from one of the first and second clock domains to the other of the first and second clock domains at the speed matching module.

45. (Previously Presented) The method of claim 41 wherein the first and second processors share memory mapped registers within the servo controller.

46. (Previously Presented) The method of claim 41 further comprising resolving transaction conflicts between the first processor and the second processor at a pipeline control module.

47. (Previously Presented) The method of claim 41 further comprising delaying a write access for one of the first and second processors during write conflicts between the first and second processors.